

APPENDIX XII

ProtocolCheck(P) (* look at Protocol type in packet P to see if P is to be Bridged or Routed*)

ClientLink(L) (* is link L a link to a client *)

Return true if L is in ClientLinks and false otherwise;

AllBridgeLinks

Return set of ServerLinks that are turned on in Spanning Tree excluding links in ClientLinks

FindClientLinks(V,L) (* find client links other than L for Vlan V *)

Let VR be the VlanRecord associated with VlanId V

If L is not in VR.ClientLinks or VR.VlanLinks then Return(EmptyList)

Else

Return (VR.ClientLinks - L) (* return VR.ClientLinks after removing L if L is in list*)

FindVlanLinks(V,L) (* find Vlan links other than L for Vlan V *)

Let VR be the VlanRecord associated with VlanId V

If L is not in VR.ClientLinks or VR.VlanLinks then Return(EmptyList)

Else

Return (VR.VlanLinks - L) (* return VR.VlanLinks after removing L if L is in list *)

VlanfromLink(L); (* returns Vlan that link L belong to for routing *)

For each VlanRecord VR in VlanList do

If L is in VR.Links and VR.Type = IntegratedRouting then Return(VR.VlanId)

VlanfromPacket(P) (* decodes Vlan field encoded in a packet *)

(* returns decoding of VlanId as described at end of Chapter 6*)

Forward (P, LinkSet) (* Forward packet P to links specified in LinkSet *)

AddVlanId(P,V)

P = Encode(P,V)

P = AddCRC(P)

(* add encoding of VlanId to a packet P *)

(* encoding function is listed at end of Chapter 6 *)

(* recalculate CRC for P *)

RemoveVlanId(P)

P = RemoveCode(P)

P = AddCRC(P)

(* remove encoding of VlanId from a packet P *)

(* function to remove VlanID code is listed at end of Chapter 6 *)

(* recalculate CRC for P *)